Graduate Seminar on Discrete Optimization

Summer 2023

Maximum Flow and Minimum-Cost Flow in Almost-Linear Time

The seminar will be based on the paper [3]. Before the first talk, every participant should watch the talk by Rasmus Kyng available here: https://www.youtube.com/watch?v=87mhWeEjiec. Moreover, every participant should have read the introduction (Section 1) of [3].

Section 2 of [3] provides an overview of the algorithm and Section 3 contains some preliminaries. Participants should read and present material from these sections whenever this is helpful for explaining or motivating the content of their assigned section.

Topic ① provides a proof of Informal Theorem 1.4 from [3], Topics ② and ③ provide a proof of Informal Theorem 1.6 from [3], and Topics ④ – ⑧ provide a proof of Informal Theorem 1.5 from [3].

① **Potential Reduction Interior Point Method (1 talk)**

Sections 2.1 and 4.1–4.2 of [3].

② **Expander Decompositions (1 talk)**

Appendix B.2 of [3] and [6].
The following talk about [6] might also be helpful: https://www.youtube.com/watch?v=Q8hxG11zVdc

③ **Decremental Spanners with Embedding**

See Section 2.7 from [3].

3a **The Algorithm (1 talk)**

Theorem 5.1 and Section 5.1 from [3].

3b **Implementing the Sparsification procedure (1 talk)**

Section 5.2 from [3], including the proof of Theorem (Section 3.2 from [4]).

④ **Low Strech Spanning Trees (1 talk)**

Theorem 3.2 from [3] and its proof in [1].
See also "Warm-Up: A Simple, Static Algorithm" in Section 2.2.

⑤ **Link-Cut Trees (1 talk)**

Lemma 3.3 and Lemma B.8 from [3] and their proofs from [5].
See also Chapter 5 of [7].

⑥ **Data Structure Chain (1 talk)**

Sections 2.2 – 2.3 and Section 6 from [3].
See also Appendix B.3 and B.4 from [3] for more details.

(7) **Routing and Cycle Quality Bounds (2 talks)**

Sections 2.4–2.5 and 7 from [3].

(8) **Rebuilding Data Structure Levels (1 talk)**

Sections 2.6 and 8 from [3].

(9) **The Overall Algorithm (1 talk)**

Section 9 from [3] and Section 4.3 (including the proofs of Lemma 4.11 and 4.12, which can be found in [2] and Appendix B.1 from [3]).

References

[1] Ittai Abraham and Ofer Neiman. "Using petal-decompositions to build a low stretch spanning tree". In: *SIAM Journal on Computing* 48.2 (2019), pp. 227– 248.

[2] Jan van den Brand, Yin Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, Di Wang. "Bipartite matching in nearly-linear time on moderately dense graphs". In: *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pp. 919–930. https://arxiv.org/pdf/2009.01802.pdf

[3] Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time. In: *Proceedings of the 63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pp. 612–623. https://arxiv.org/abs/2203.00671v2

[4] Julia Chuzhoy and Thatchaphol Saranurak. "Deterministic algorithms for decremental shortest paths via layered core decomposition". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*, pp. 2478–249. https://arxiv.org/abs/2009.08479v1

[5] Daniel D. Sleator and Robert Endre Tarjan. "A data structure for dynamic trees" In: *Journal of computer and system sciences* 26.3 (1983), pp. 362–391.

[6] Thatchaphol Saranurak and Di Wang. "Expander Decomposition and Pruning: Faster, Stronger, and Simpler". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pp. 2616–2635. https://arxiv.org/abs/1812.08958

[7] Robert Endre Tarjan. "Data Structures and Network Algorithms". Society for Industrial and Applied Mathematics, 1983.